

TECNOLOGÍAS ASOCIADAS A LAS APLICACIONES WEB

Existen varios tipos de tecnologías para los Servidores Web, estas tecnologías se pueden dividir en 4 grupos principales que son:

- Tecnologías al lado del cliente
- Tecnologías del documento
- Tecnologías al lado del servidor
- Tecnologías para los servicios web

Dentro de las Tecnologías al lado del cliente se encuentran las siguientes:

1. Navegadores
2. Scripts de JavaScript, Scripts de VisualBasic
3. Plug-ins Flash
4. Applets Java, limitados

En las tecnologías del documento nos encontramos las siguientes:

1. HTML, XHTML
2. XML, DTD, XML SCHEMA, SML DOM, X-PATH
3. CSS, XSL, XSLT, XSL-FO
4. SVG
5. SMIL
6. VOICEML
7. WML

Como tecnologías al lado del servidor podemos ver:

1. Módulos compilados
2. Scripts
3. Programación distribuida RMI, IIOP
4. ASP.NET

Por último como tecnologías para los servicios web tenemos los siguientes, los cuales pasamos a desarrollar más detenidamente.

SOAP

SOAP => **S**imple **O**bject **A**ccess **P**rotocol = (Protocol de acceso a objetos simples)

Soap es un protocolo estándar basado en http que permite que dos objetos en diferentes procesos puedan comunicarse a través de un intercambio de datos en XML. Soap permiten que diferentes aplicaciones ejecutadas en diferentes sistemas operativos, con diferentes tecnologías y lenguajes de programación distintos intercambien datos entre ellas. Fue creado por Microsoft, IBM y otros.

Un mensaje SOAP es un documento XML normal y corriente que contiene los siguientes elementos:

- Una etiqueta Envelope que identifica el documento XML como un mensaje SOAP.
- Un elemento Header que contiene la información del encabezado.
- Un elemento body que recoge la información del cuerpo.
- Un elemento Fault que contiene errores y el estado de la información.

Ejemplo de mensaje SOAP:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
  ...
</soap:Header>

  <soap:Body>
  ...
    <soap:Fault>
    ...
    </soap:Fault>
  </soap:Body>

</soap:Envelope>
```

Reglas de la sintaxis de un mensaje SOAP:

- El mensaje SOAP debe estar codificado utilizando XML.
- Debe tener la etiqueta Envelope.
- Debe tener especificado la codificación.
(<http://www.w3.org/2001/12/soap-encoding>)
- No debe contener ninguna referencia DTD.

- No debe contener las instrucciones de procesamiento de XML.

Un ejemplo real de mensaje SOAP:

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

```
<?xml version="1.0"?>
```

```
<soap:Envelope
```

```
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
```

```
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

```
  <soap:Body xmlns:m="http://www.example.org/stock">
```

```
    <m:GetStockPrice>
```

```
      <m:StockName>IBM</m:StockName>
```

```
    </m:GetStockPrice>
```

```
  </soap:Body>
```

```
</soap:Envelope>
```

Mediante este mensaje SOAP se le pide al host www.example.org que devuelva el Precio de stock del product IBM.

WSDL

WSDL = Web Services Description Language (Lenguaje descriptivo de los servicios web)

Igual que SOAP es un documento escrito en XML. Especifica la localización de los servicios y los métodos que el servicio ofrece.

Elementos de la estructura de un documento WSDL:

- <types> esta sección define los tipos de datos usados en los mensajes.
- <message> en esta sección se definen los elementos del mensaje.
- <portType> Se definen las operaciones permitidas.
- <binding> Se especifican los protocolos utilizados en la comunicación.

Ejemplo:

```
<message name="getTermRequest">  
  <part name="term" type="xs:string"/>  
</message>
```

```
<message name="getTermResponse">  
  <part name="value" type="xs:string"/>  
</message>
```

```
<portType name="glossaryTerms">  
  <operation name="getTerm">  
    <input message="getTermRequest"/>  
    <output message="getTermResponse"/>  
  </operation>  
</portType>
```

REST

La **Transferencia de Estado Representacional** o Representational State Transfer, REST, es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.

El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding

Si bien el término REST se refería originalmente a un conjunto de principios de arquitectura, en la actualidad se usa en el sentido más amplio para describir cualquier interfaz web simple que utiliza XML y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como el protocolo de servicios web SOAP.

Es posible diseñar sistemas de servicios web de acuerdo con el estilo arquitectural REST de Fielding y también es posible diseñar interfaces XMLHTTP de acuerdo con el estilo de llamada a procedimiento remoto pero sin usar SOAP.

REST afirma que la web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:

- Un **protocolo cliente/servidor sin estado**: cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST)
- Un **conjunto de operaciones bien definidas** que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de operaciones, las más importantes son **POST, GET, PUT y DELETE**. Con frecuencia estas operaciones se equiparan a las operaciones CRUD que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.
- Una **sintaxis universal** para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- El **uso de hipermedios**, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente **HTML o XML**. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

Un concepto importante en REST es la existencia de recursos, que pueden ser accedidos utilizando un identificador global. Para manipular estos recursos, los componentes de la red se comunican a través de una interfaz estándar, HTTP, e intercambian representaciones de estos recursos

La petición puede ser transmitida por cualquier número de conectores, pero cada uno lo hace sin "ver más allá" de su propia petición, esto se conoce como separación en capas.

Así, una aplicación puede interactuar con un recurso conociendo el identificador del recurso y la acción requerida, no necesitando conocer si existen cachés, proxys, cortafuegos, túneles o cualquier otra cosa entre ella y el servidor que guarda la información.

La aplicación, sin embargo, debe comprender el formato de la información devuelta, que es por lo general un documento HTML o XML, aunque también puede ser una imagen o cualquier otro contenido.

UDDI

UDDI son las siglas del catálogo de negocios de Internet denominado Universal Description, Discovery and Integration.

El registro en el catálogo se hace en XML.

UDDI es una iniciativa industrial abierta entroncada en el contexto de los servicios Web.

El registro de un negocio en UDDI tiene tres partes:

- Páginas blancas - dirección, contacto y otros identificadores conocidos.
- Páginas amarillas - categorización industrial basada en taxonomías.
- Páginas verdes - información técnica sobre los servicios que aportan las propias empresas.

UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.